SPM

Project life cycle

**Software testing**

**Stage of testing**

1. Unit test
2. Integration testing
3. System testing
4. System integration testing
5. User Acceptance Test (UAT)

**Type of test**

1. Functional

* Function of the software e.g. search, sort

1. Non-functional

* Performance testing (able to handle large amount of data)
* Usability testing (easy to use)

1. Black-box testing

* Tester doesn’t know the inner code test on functionality

1. White-box testing

* Tester know the inner code, test on code level (Katalon studio)

1. Grey-box testing

* Mix of black and white box

1. Regression testing

* Test of function in both current iteration and previous iteration to make sure the new code doesn’t cause the old code to crash

**UAT**

1. Alpha testing

* By the developer

1. Beta testing

* By the client/ user

1. Requirement for UAT
2. Functionality requirement
3. Performance requirement
4. Interface requirement
5. Overall quality requirement

**Test case**

* The summary of scenarios for a single task
* Description
* Input
* Result

Area | Description | Input | Test procedure |Expected result |Result | Pass/Fail

**Software processes**

**Waterfall process**

Assume the plan is clear from the start

**Process**

1. Requirement
2. Define
3. Code
4. Integrate
5. Test
6. Deployment
7. Maintenance

**Iterative process**

Earlier iteration should address the greatest risk

Can handle uncertainty

**Agile process**

Iterative + incremental process

Focus on

Individual & interaction over processed tool

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

**Rational unified process (RUP)**

* A software engineering process framework to configure your own process
* 4 phases

1. Inception – understand what to build
2. Prepare vision document and initial business case
3. Develop high level project requirement - 10-20% of the domain model and use case completed
4. Manage process – reduce risk by identify all key requirement and acknowledge requirement will change
5. Elaboration – understand how to build
6. Detail requirement 80% done
7. Develop executable and stable architect – 10% code done
8. Test architect with key use case
9. Verify architect quality
10. Continue assess business case, risk profile and development plan
11. Construction – build the product
12. Complete requirement and design model
13. Design implement and test each component
14. Test each build – automated regression testing
15. Produce fully functional software - Beta release
16. Release description
17. Transition – validate and deploy the product
18. Produce incremental “bug -fix” release
19. Update user manual and deployment documentation
20. Update release description
21. Execute cut-over

**SCRUM**

Roles

Scrum Master

* Main job is to remove impediments for the team so that they can work on the project
* Project manager role
* Responsible to enacting scrum value and practices

Product Owner

* Product manager role
* Know what needs to be build and in what sequence – give suggestion to direct and things to do

Scrum team

* Cross functional
* 5-10 people
* Self-organized
* Change of membership can be done between sprint
* Full time

Process

1. Sprint planning meeting
2. Creating product backlog  
   Participants: Scrum master, Scrum team, Product owner
3. Creating sprint backlog

Participants: Scrum master, Scrum team

1. Sprint – code, test, integrate features
2. Daily scrum – share what they work on the previous day and what they are going to do today and any impediments
3. Sprint review meeting – demo work done and get feedback

Artifacts

1. Sprint backlog – cannot be change within the sprint
2. Product backlog – changes can add to product backlog

**Project management**

****

**Step 1: define scope**

Project scope identify & define tasks to meet the project objective

Tasks are based on the requirements from the key user

1. Functionality
2. Usability
3. Reliability
4. Performance
5. Supportability

**Step 2: decide process**

based on strength of the process

1. Project size
2. Team distribution
3. Complexity of technology
4. Number of stakeholders
5. Compliance requirements
6. Where in the project life cycle

**Step 3: define work breakdown structure**

****

Break down until a single work unit

**Step 4: create schedule**

1. Define activity
2. Sequence activities
3. Allocation resource
4. Estimate duration
5. Develop schedule

**Step 5: Estimate cost**

* Determine resource cost rate
* Vendor bid analysis
* Cost of quality

**Step 6: identify risk**

* Risk avoidance: reorganized the project plan to eliminate risk
* Risk mitigation: reduce the probability or impact of the risk to an acceptable level
* Risk transfer: outsource a third party to take the risk
* Risk acceptance: accept the risk
* Contingency plan: plan B

**Bad scheduling**

* **Estimates are never updated and improved from the previous iteration**
* **Tasks that are of too long a duration (15 days)**
* **Tasks that are too general (Task Coding)**
* **Tasks that has nobody assigned to It** 
  + **or the "team" is assigned to it**
  + **Variant: small set of people assigned most of the tasks**
* **Same tasks for all iterations / Generic Tasks**
  + **Nobody knows what is suppose to be done.**
  + **"code", "update use case" etc**
* **Schedule done on a week basis / not a task basis**
  + **Week 1 : stuff, Week 2 : same stuff**
  + **You are not hourly workers!!**

**Quality management**

Goal question metric (GQM)

1. What to improve
2. What detail to ask about thep rocess
3. What number can be put on the question